

Flipkart	Flipkart
Features/Functionalities - Login, search, add to cart, payment>debit card, credit card, upi etc, header section	Features/Functionalities - Login, search, add to cart, payment>debit card, credit card, upi etc, header section
1) To verify login functionality(main scenario) >	1) To verify login functionality
To verify login functionality by entering valid credentials(sub scenario)	2) To verify search functionality
To verify login functionality by entering invalid credentials(sub scenario)	3) To verify add to cart functionality
	4) To verify payment functionality

OpenCart-Test Scenarios [XLSX](#)

File Edit View Insert Format Data Tools Help

View only

C5

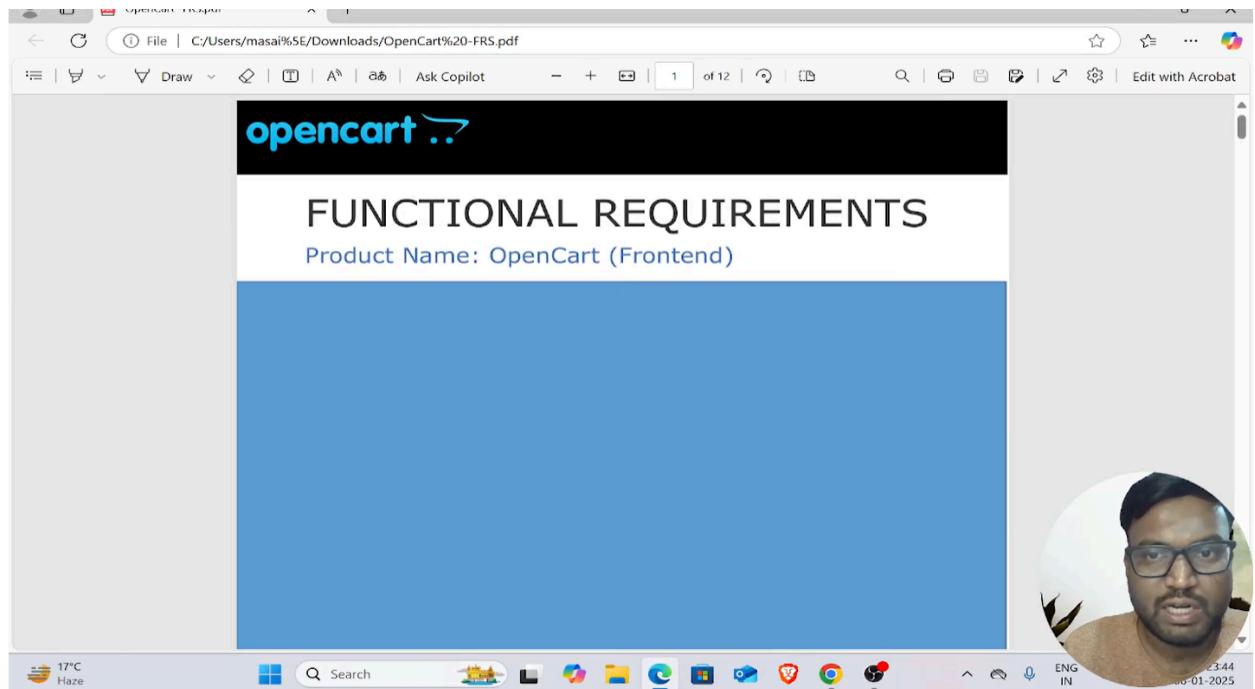
Pavan

A	B	C	D	E
Project Name	OpenCart (Frontend)			
Client	OpenCart			
Reference Document	FRS			
Created By	Pavan			
Creation Date	DD-MM-YYYY			
Approval Date	DD-MM-YYYY			
Test Scenario ID	Reference	Test Scenario Description	Priority	Number of Test Cases
TS_001	FRS	Validate the working of Register Account functionality	P0	27
TS_002	FRS	Validate the working of Login functionality	P0	23
TS_003	FRS	Validate the working of Logout functionality	P0	11
TS_004	FRS	Validate the working of Forgot Password functionality	P2	25
TS_005	FRS	Validate the working of Search functionality	P1	22
TS_006	FRS	Validate the working of Product Compare functionality	P4	24
TS_007	FRS	Validate the Product Display Page functionality for the different types of Products	P1	37
TS_008	FRS	Validate the working of 'Add to Cart' functionality	P1	09
TS_009	FRS	Validate the working of 'Wish List' functionality	P4	21
TS_010	FRS	Validate the working of 'Shopping Cart' functionality	P1	33
TS_011	FRS	Validate the working of Home Page functionality	P2	10
TS_012	FRS	Validate the working of Checkout functionality	P1	20
TS_013	FRS	Validate the My Account functionality	P2	9
TS_014	FRS	Validate the working of My Account > Account Information functionality	P3	13
TS_015	FRS	Validate the working of My Account > 'Change Password' functionality	P3	13
TS_016	FRS	Validate the working of My Account > 'Address Book' functionality	P3	21
27	TS_017	Validate the working of My Orders > 'Order History' functionality	P3	12
28	TS_018	Validate the working of My Orders > 'Order Information' functionality	P3	8
29	TS_019	Validate the working of My Orders > 'Product Returns' functionality	P3	11
30	TS_020	Validate the working of My Orders > 'Downloads' functionality	P3	13
31	TS_021	Validate the working of My Orders > 'Reward Points' functionality	P3	10
32	TS_022	Validate the working of My Orders > 'Returned Requests' functionality	P3	17
33	TS_023	Validate the working of My Orders > 'Your Transactions' functionality	P3	11
34	TS_024	Validate the working of My Orders > 'Recurring Payments' functionality	P3	9
35	TS_025	Validate the working of 'Affiliate' functionality	P4	29
36	TS_026	Validate the working of 'Newsletter' functionality	P4	13
37	TS_027	Validate the working of 'Contact Us' page functionality	P4	13
38	TS_028	Validate the working of 'Gift Certificate' page functionality	P4	11
39	TS_029	Validate the working of 'Special Offers' page functionality	P4	16

Industries follows To verify/To validate/To check most often.

Here Reference Document is FRS. By looking into FRS tester needs to create test Scenarios.

Document look like below image:



Prepared by: Pavan
Date: Jan 10, 2021

Overview

OpenCart is free open source e-commerce platform for online merchants. OpenCart provides a professional and reliable foundation from which to build a successful online store. This foundation appeals to a wide variety of users; ranging from seasoned web developers looking for a user-friendly interface to use, to shop owners just launching their business online for the first time. OpenCart has an extensive amount of features that gives you a strong hold over the customization of your store. With OpenCart's tools, you can help your online shop live up to its fullest potential.



Requirements

OpenCart requires certain technical requirements to be met for the store to operate properly. First, a web server must be created to make the OpenCart store publicly available on the web. Domain names and hosting services can easily be purchased for an affordable price.

When selecting a hosting service, you should check to see that these server requirements are provided and installed on their web servers:

These extensions must be enabled for OpenCart to install properly on the web server.

- PHP 5.4
- jQuery 2.1.1
- JavaScript
- Database (MySQL suggested)
- Web Server (Apache suggested)

Opencart Frontend

This Document is intended to be used as an introduction to the OpenCart default store front. The store front reveals how the customer views and interacts with the store.

The screenshot shows a standard OpenCart store interface. At the top, there's a navigation bar with links for 'Currency', '123456789', 'My Account', 'Wish List (0)', 'Shopping Cart', and 'Checkout'. Below the navigation is a search bar and a user account section. The main header 'Your Store' is displayed above a menu bar containing categories like 'Desktops', 'Laptops & Notebooks', 'Components', 'Tablets', 'Software', 'Phones & PDAs', 'Cameras', and 'MP3 Players'. The main content area features a large image of two laptops, one standing upright and one lying flat, both displaying a photo of a couple. Below this is a 'Featured' section with four product cards: 'MacBook' (laptop), 'iPhone' (mobile phone), 'Apple Cinema 30'' (monitor), and 'Canon EOS 5D' (camera). Each card includes a small image, the product name, a brief description, and price information.

Featured

Product	Description	Price
MacBook	Intel Core 2 Duo processor Powered by an Intel Core 2 Duo processor at speeds up to 2.1..	\$602.00
iPhone	iPhone is a revolutionary new mobile phone that allows you to make a call by simply tapping a name..	\$123.20
Apple Cinema 30"	The 30-inch Apple Cinema HD Display delivers an amazing 2360 x 1600 pixel resolution. Designed sp...	\$110.00 - \$122.00
Canon EOS 5D	Canon's press material for the EOS 50 states that it 'defines (a) new D-SLR category', while we'r...	\$98.00 - \$122.00

FRS documents is done before actual building of the software. The above image shows the mock screens or the dummy picture that shows how the product looks like before building the actual product.

The screenshot displays a sample OpenCart store interface. At the top, there are four product cards with the following details:

- Product 1: \$602.00 (Ex Tax: \$400.00) - Add to Cart
- Product 2: \$125.20 (Ex Tax: \$101.00) - Add to Cart
- Product 3: \$110.00 (Ex Tax: \$90.00) - Add to Cart
- Product 4: \$98.00 (Ex Tax: \$80.00) - Add to Cart

Below the products are five brand logos: DELL, Disney, STARBUCKS COFFEE, Nintendo, and NFL.

The footer area contains navigation links:

- Information:** About Us, Delivery Information, Privacy Policy, Terms & Conditions.
- Customer Service:** Contact Us, Returns, See Map.
- Extras:** Brands, Gift Certificates, Affiliate, Specials.
- My Account:** My Account, Order History, Wish List, Newsletter.

Powered By OpenCart Version 3.0.2.0 © 2018

The products seen above are included as sample data with the OpenCart installation. These products can easily be removed and replaced with the shop's products later.



Navigating the shop

The OpenCart default theme makes navigating a shop's products easily accessible to its customers.

Home page

The home page is arguably the most important page in the shop, in terms of presentation. In most cases, this will be the first page that a customer interacts with (especially if they are directed to the store site from a search engine). **The shop's homepage needs to be user-friendly, while at the same time highlighting the shop's products.**

The first step in becoming familiar with the store front is understanding the anatomy of the OpenCart default homepage.

The header

The header will be displayed at the top of the page, on every page of the store; not just the home page.



The header has the following navigation options:

The screenshot shows a product page for an HP laptop. At the top, there's a navigation bar with links for Desktops, Laptops & Notebooks, Components, Tablets, Software, Phones & PDAs, Cameras, and MP3 Players. Below the navigation is a breadcrumb trail showing the category path: Home > Brand > Hewlett-Packard. The main content area features a large image of the laptop, its title 'HP LP3065', and a short description: 'Stop your co-workers in their tracks with the stunning new...'. To the right of the product image is a circular profile picture of a man with glasses.

The top header is still common after selecting the product. Not just in the home page. If it is not showcasing the top header then it is marked as the defect because it is not satisfying the FRS document.

The header has the following navigation options:

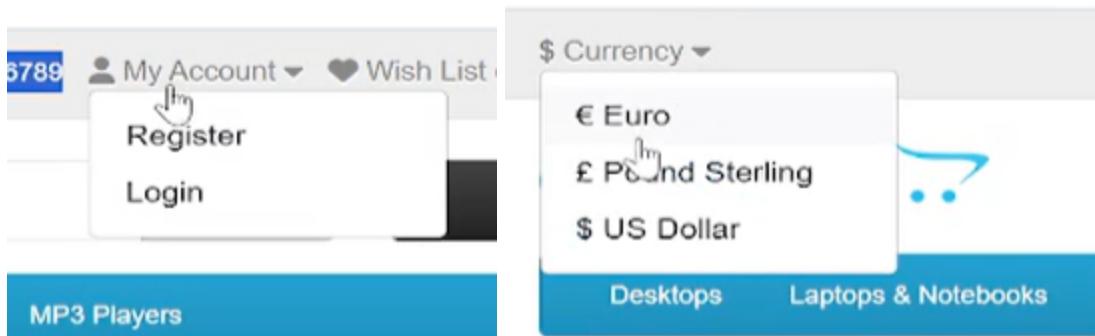
- **Store logo:** Clicking on this logo will direct the customer back to the home page of the store.
- **Currency block:** The customer can select which currency the store's products will be in by clicking on any of the currency icons.
- **Shopping Cart:** Displays the number of items purchased, and the total price of the order. Clicking on the button will contain all of products added to the cart and an option to "View Cart" or "Checkout".
- **Search box:** The customers can type in the search box to search for a product within the store's product categories.
- **Links:** Links the customer to the Home page, Wish List, My Account, Shopping Cart, and Checkout.

By click on the logo it must take to the home page otherwise defect.

- **Telephone:** Company telephone number.
- **My Account:** Customer can register or login from here.

The top menu

The top menu category only displays the top parent categories of products.



The top menu

The top menu category only displays the top parent categories of products.

WWW.PAYANONLINETRAININGS.COM



Like the header, the top menu will be displayed on every page. When the customer's mouse is dragged over a category, a drop-down menu will display the sub-categories for that parent category.

Desktops Laptops & Notebooks Components Tablets Software Phones & PDAs Cameras MP3 Players

When a parent category is clicked, the customer will be directed to the category page, which displays all the products within that category.

Slideshow

The slideshow displays several product banners of your choice by alternating the images in a slideshow. After a certain amount of time, one banner will shift to the next banner. Banners in this slideshow are

Slideshow

The slideshow displays several product banners of your choice by alternating the images in a slideshow. After a certain amount of time, one banner will shift to the next banner. Banners in this slideshow are useful for highlighting certain products to be easily accessible by the customer. When the banner is clicked on, the customer will be directed to the product on the banner's page.



Unlike the top menu and header, the slideshow in the OpenCart default can only be viewed on the home page in this position.

Featured products

OpenCart gives you the option of featuring specific products of their choosing on the home page.



Featured

MacBook

Intel Core 2 Duo processor Powered by an Intel Core 2 Duo processor at speeds up to 2.1..

\$602.00
Ex Tax: \$500.00

[ADD TO CART](#) [Heart](#) [Compare](#)

iPhone

iPhone is a revolutionary new mobile phone that allows you to make a call by simply tapping a name.

\$123.20
Ex Tax: \$101.00

[ADD TO CART](#) [Heart](#) [Compare](#)

Apple Cinema 30"

The 30 inch Apple Cinema HD Display delivers an amazing 2560 x 1600 pixel resolution. Designed sp...

\$110.00 **\$120.00**
Ex Tax: \$90.00

[ADD TO CART](#) [Heart](#) [Compare](#)

Canon EOS 5D

Canon's press material for the EOS 5D states that it 'defines a new D-SLR category', while we...

\$98.00 **\$122.00**
Ex Tax: \$80.00

[ADD TO CART](#) [Heart](#) [Compare](#)

The Featured section includes the product image, name, price and an option to add the product directly to the Shopping Cart.

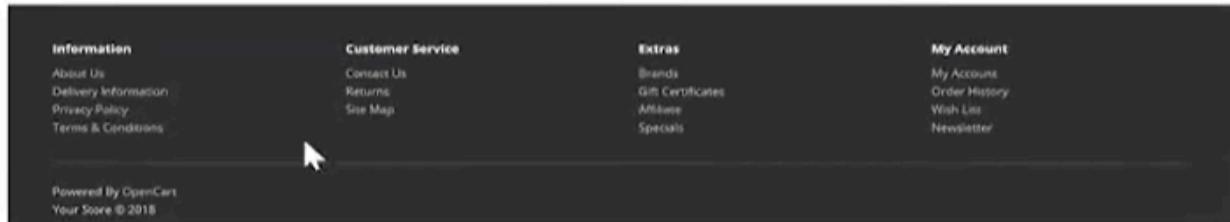
The carousel is only located on the Home Page in the default.

Footer

The footer is located at the bottom of every page, not just the Home Page. This block of miscellaneous links is useful in sorting relevant pages for the customer that may not logically sort anywhere else.

The organizational scheme of the footer can be divided into the following sections:

- **Information:** "About Us", "Delivery Information", "Privacy Policy", "Terms & Conditions"
- **Customer Service:** "Contact Us", "Returns", "Site Map"
- **Extras:** "Brands", "Gift Vouchers", "Affiliates", "Specials"
- **My Account:** "My Account", "Order History", "Wish List", "Newsletter"



Product pages

The OpenCart default product page will follow the structural format seen below.

Desktops Laptops & Notebooks Components Tablets Software Phones & PDAs Cameras MP3 Players

MacBook



MacBook

Brand: Apple
Product Code: Product 16
Reward Points: 600
Availability: In Stock

\$602.00

Ex Tax: \$500.00

Qty:

 0 reviews / Write a review

 Like 12  Tweet  Share 1.3K

Description Specification Reviews (0)

Intel Core 2 Duo processor
Powered by an Intel Core 2 Duo processor at speeds up to 2.16GHz, the new MacBook is the fastest ever.

1GB memory, larger hard drives
The new MacBook now comes with 1GB of memory standard and larger hard drives for the entire line perfect for running more of your favorite applications and storing growing media collections.

Sleek, 1.08-inch-thin design
MacBook makes it easy to hit the road thanks to its tough polycarbonate case, built-in wireless technologies, and innovative MagSafe Power Adapter that releases automatically if someone accidentally trips on the cord.

Built-in iSight camera
Right out of the box, you can have a video chat with friends or family, record a video at your desk, or take fun pictures with Photo Booth.

The product page can be divided into the following sections:

- **Product image:** The product image can be displayed under the title on the left-side, along with alternate views of the product underneath it in smaller box. Clicking on the main image will expand the image within the window for the customer to see it in greater detail.
- **Product details:** The product code, availability, and price are displayed just right to the product image.
 - **Cart:** The customer can select a quantity and add the product to their cart, wish list, or compare.
 - **Rating/Sharing:** Underneath the cart can rate the product and/or share the product on different social media websites.
 - **Description tab:** An area underneath the main product information to provide a detailed description of the product.
 - **Review tab:** An area for the customer to write a review on the product.

Category product listings

If the product image is not expanding after clicking it then its a defect.

Certainly! Here's an **interview-style explanation** of **FRS (Functional Requirement Specification)** and **SRS (Software Requirement Specification)** in **manual testing**, including **use cases**, **differences**, and **why they matter**.

 **Question: What is FRS and SRS in Manual Testing? Explain with use cases.**

 **Answer:**

♦ **FRS (Functional Requirements Specification):**

FRS is a document that outlines the **functional requirements** of a software application. It focuses on **what the system should do** — the features, behaviors, and interactions of the software from a **user's perspective**.

Key Points:

- Describes business needs in terms of software features.
- Written in simple, business-friendly language.
- Includes use cases, user flows, and UI behavior.

Use Case Example:

For an **e-commerce app**, an FRS might describe:

"The user should be able to add products to the cart, view the cart, and proceed to checkout with valid payment options."

♦ **SRS (Software Requirements Specification):**

SRS is a more **detailed technical document** that contains both **functional and non-functional requirements**. It's used by developers, testers, and project managers to understand **how** the system must be built.

Key Points:

- Combines **FRS + Non-Functional Requirements (NFR)** like performance, security, scalability.

- Serves as a base for test planning, system design, and development.
- May include diagrams, data models, and technical constraints.

Use Case Example:

For the same e-commerce app, the SRS might specify:

- "When a user clicks 'Add to Cart', an API call should be triggered with the product ID and user session token."
- "The response time for the API should be less than 2 seconds."



In Manual Testing – Why Are FRS and SRS Important?

Role	Use of FRS/SRS
Manual Tester	Creates test cases based on FRS/SRS. Ensures all functionalities are verified as per business and system expectations.
Test Lead	Designs Test Plan , identifies test scope , and prioritizes features based on FRS/SRS.
Developer	Uses SRS to understand exact implementation details.
Client/Stakeholder	Uses FRS to validate if business needs are captured correctly.



FRS vs SRS – Key Differences

Feature	FRS	SRS
Focus	User/business functionality	System functionality (technical)
Audience	Clients, Business Analysts	Developers, Testers, Architects
Content	What the system should do	What + How system should behave
Includes	Functional requirements only	Functional + Non-functional reqs
Language	Simple, business-oriented	Technical, detailed

Summary for Interviews:

"FRS defines *what* functionalities the software should offer from a user's point of view, while SRS explains *how* those features will be technically implemented. In manual testing, both documents are essential for designing test cases, validating requirements, and ensuring the system aligns with both business needs and technical constraints."

Let me know if you want a **sample FRS/SRS document** or a **real-world scenario walkthrough!**

Test Design Techniques:

To systematically derive effective and efficient test cases.
Helps in finding defects early by identifying key testing areas.
Reduces redundancy and improves testing focus.

Test Design Techniques:

Structured approaches for creating optimized test cases.

Types include:

Black-Box Techniques: Focus on functionality (e.g., Equivalence Partitioning, Boundary Value Analysis).

White-Box Techniques: Focus on code structure (e.g., Statement Coverage, Branch Coverage).

Experience-Based Techniques: Leverage intuition and domain knowledge (e.g., Exploratory Testing).

Equivalence Partitioning



Equivalence Partitioning is a software testing technique used in manual testing to systematically divide the input data of a system into groups or partitions that are expected to exhibit similar behavior. The goal is to test representative values from each partition to ensure that the software functions correctly and to optimize test coverage.

Examples of Equivalence Partitioning:

Example 1: Input Field Accepting Numbers from 1 to 100

Valid Partition: 1 to 100

Test case: Input = 50

Invalid Partition:

Below the range: Test case: Input = 0

Above the range: Test case: Input = 101

Non-numeric input: Test case: Input = "abc"

Error Guessing

Error Guessing

- no rules, no script
- situations in which the software may not be able to cope are identified based on experience(feel)
- Consider:
 - past failures
 - brain storming
 - intuition
 - "what is the craziest thing we can do"



Author: Shashank Chauhan

10/10/15

Equivalence Class Partitioning

- 1) Need to test the software if it accepts numbers in the range of 1 to 100

0 to -10 = -5
1 to 10, ~~✓~~ 4
11 to 20, = 12
21 to 30, = 25
31 to 40, = 33
41 to 50=45
51 to 60=56
61 to 70=64
71 to 80=81
81 to 90=88
91 - 100=94
101 to 110=106

Assume you are having full knowledge on manual testing and a good guide to explain to the beginner the provide best examples from beginner to advance for the equivalence partitioning at first provide definition, its comes under what category.

Certainly, Routh! Let's break down **Equivalence Partitioning** in a structured and beginner-friendly way, and then move step-by-step toward advanced examples.

Definition of Equivalence Partitioning

Equivalence Partitioning (EP) is a **black box testing technique** used in **manual testing** where input data is divided into **partitions or classes** of equivalent data from which test cases can be derived.

The idea is:

- **If one value in a partition works**, all other values in that partition are assumed to behave similarly.
 - Instead of testing all possible inputs, we test **just one value from each partition**, saving time and effort while maintaining test coverage.
-

Category:

- **Type:** Black Box Testing Technique
 - **Belongs to:** Test Case Design Techniques
 - **Goal:** Reduce number of test cases while ensuring effective test coverage.
-

Basic Example (Beginner Level)

Scenario:

A user must enter their age in a form. The valid age range is **18 to 60** (inclusive).

Partitions:

1. **Valid Partition:** 18 to 60 → (Expected to pass)
2. **Invalid Partition 1:** Less than 18 → (Expected to fail)
3. **Invalid Partition 2:** Greater than 60 → (Expected to fail)

Test Cases using EP:

Test Case	Input	Expected Result
TC1	25	Accepted
TC2	15	Rejected
TC3	65	Rejected

→ You pick one value from each partition.

 Intermediate Example**Scenario:**

A mobile app accepts passwords that must be:

- 8 to 12 characters long
- Only alphanumeric characters

Equivalence Partitions:

1. Valid length and characters →
2. Too short (<8 characters) →
3. Too long (>12 characters) →

4. Contains special characters →

Test Cases:

Test Case	Input	Expected Result
TC1	abc12345	Accepted
TC2	ab12	Rejected
TC3	abc123456789 0	Rejected
TC4	abc@1234	Rejected

Advanced Example

Scenario:

An online banking system allows transfer amounts:

- Minimum: ₹100
- Maximum: ₹1,00,000
- Must be multiples of ₹100

Partitions:

1. **Valid:** ₹100 – ₹1,00,000 AND divisible by 100 →
2. **Invalid 1:** Less than ₹100 →
3. **Invalid 2:** Greater than ₹1,00,000 →
4. **Invalid 3:** Not divisible by ₹100 →

Test Cases:

Test Case	Input	Expected Result
-----------	-------	-----------------

TC1	₹5000	Accepted
TC2	₹50	Rejected
TC3	₹15000 0	Rejected
TC4	₹550	Accepted
TC5	₹555	Rejected

→ Although ₹555 is between 100 and 100000, it's invalid because it's not a multiple of 100.

⌚ Key Takeaways

- Reduces test cases by testing representative values.
 - Helps identify **invalid input boundaries** quickly.
 - Efficient and **time-saving** compared to testing every single input.
-

Would you like me to provide a **comparison between Equivalence Partitioning and Boundary Value Analysis (BVA)** next or create **real-world test case tables for a website or app using EP?**

Boundary Value Analysis

Boundary Value Analysis (BVA) is a black-box testing technique that focuses on testing the boundaries of input values for a given system. It is widely used in software testing to ensure that the application functions correctly at the edges of the input domain.



Boundary Value Analysis

Here, we test the boundary conditions

username text field accepts min 10 & max 20 characters



To verify username text field accepts 9 characters

To verify username text field accepts 10 characters

if else condition

$\geq \leq$

when we have 2 values A & B,
we test A-1, A, A+1, & B-1, B, B+1

Key Concepts:**Boundary Values:**

Lower Bound: The smallest valid input value.

Upper Bound: The largest valid input value.

Edge Values: The values just above and below the boundaries.

Test Cases:

BVA involves testing at the boundaries and just beyond them to ensure robustness.

Test cases are designed for values at the lower bound, upper bound, and edges.

Example:

If a system accepts input values between 1 and 100, the boundary values would be 1, 100, 0 (just below the lower bound), and 101 (just above the upper bound).

Advantages:

Efficient at catching errors around boundaries.

Reduces the likelihood of defects in critical areas of the application.

Limitations:

Does not cover all possible scenarios.

May not be effective for non-numeric inputs.

Boundary Value Analysis

Certainly, Routh Kiran Babu! Let's dive into **Boundary Value Analysis (BVA)**—one of the most important techniques in manual testing. I'll explain it from **beginner to advanced level**, step by step, with real-life examples.



Definition of Boundary Value Analysis (BVA)

Boundary Value Analysis is a **Black Box Testing Technique** where test cases are designed based on the boundaries of input values. The idea is that errors tend to occur at the **edges** of input domains rather than in the center.

✓ Why Boundary?

Most defects occur at boundary values due to conditions like:

- \leq vs $<$
 - \geq vs $>$
 - Off-by-one errors
-



Category:

Boundary Value Analysis comes under:

Black Box Testing → Test Design Techniques → Equivalence Partitioning (Extension)



Beginner Level Example



Scenario:

A user can enter an age between **18 and 60** to register for a job.

- **Valid Range:** 18 to 60
- **Invalid Range:** below 18 or above 60



BVA Test Cases:

Test Case No.	Input	Description
TC01	17	Just below lower boundary (Invalid)
TC02	18	Exact lower boundary (Valid)
TC03	19	Just above lower boundary (Valid)
TC04	59	Just below upper boundary (Valid)
TC05	60	Exact upper boundary (Valid)
TC06	61	Just above upper boundary (Invalid)



Note: These test cases focus on the **edges** of valid and invalid partitions.



Intermediate Level Example



Scenario:

A password field must accept **6 to 12 characters**.

BVA Test Cases:

Test Case No.	Input	Description
TC01	5 chars	Below lower boundary (Invalid)
TC02	6 chars	Lower boundary (Valid)
TC03	7 chars	Just above lower boundary (Valid)
TC04	11 chars	Just below upper boundary (Valid)
TC05	12 chars	Upper boundary (Valid)
TC06	13 chars	Above upper boundary (Invalid)

Advanced Level Example

Scenario:

An online form requires **income** between ₹1,00,000 and ₹10,00,000 for eligibility.

BVA Test Cases:

Test Case No.	Input (₹)	Description
TC01	99,999	Just below minimum (Invalid)
TC02	1,00,000	Lower boundary (Valid)
TC03	1,00,001	Just above lower boundary (Valid)
TC04	9,99,999	Just below upper boundary (Valid)
TC05	10,00,00 0	Upper boundary (Valid)

TC06	10,00,00	Just above maximum (Invalid)
	1	

 **Tip:**

In real-time testing, combine **BVA** with **Equivalence Partitioning** and **Decision Table Testing** for more effective test coverage.

Summary

Concept	Description
Type	Black Box Testing
Category	Test Design Technique
Focus	Test boundaries of input ranges
Used For	Numeric inputs, string lengths, range limits, date ranges, etc.
Why Important?	Most defects occur at boundaries; helps catch off-by-one errors.

Would you like me to provide real-time scenarios for mobile/web applications or include boundary test case templates in Excel format?

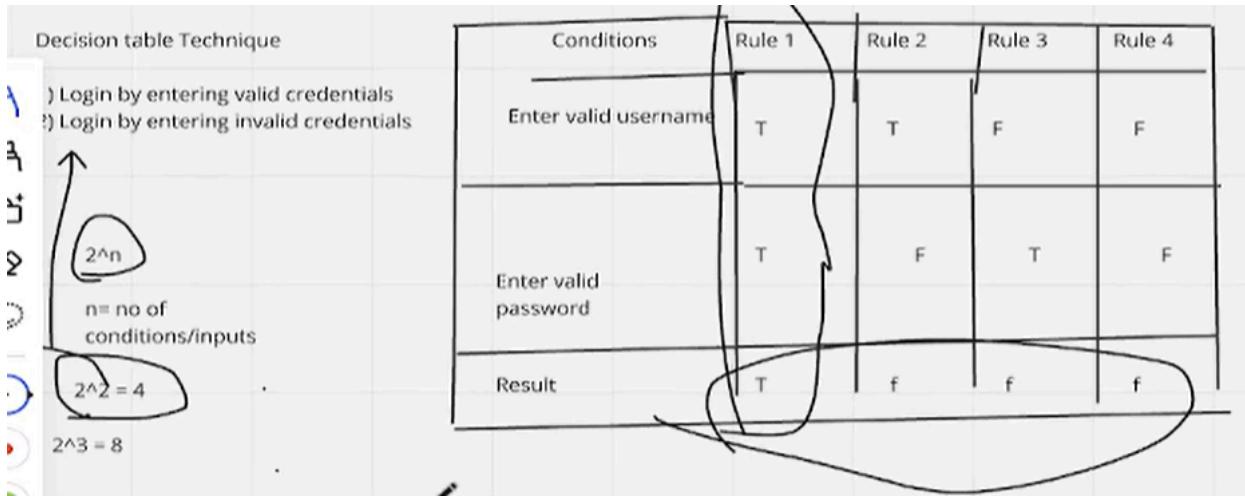
Decision Table Testing

"Decision Table Testing" is a black box test design technique in which test cases are designed to execute the combinations of inputs shown in a decision table. Decision table testing is a good way to deal with combinations of inputs.

Example 1 – How to Create a Login Screen Decision Base Table

Let's make a login screen with a decision table. A login screen with E-mail and Password Input boxes.

The condition is simple – The user will be routed to the homepage if they give the right username and password. An error warning will appear if any of the inputs are incorrect.



Certainly, Routh Kiran! Let's break down **Decision Table Analysis** clearly from **beginner to advanced**, so it's easy to understand and apply in **manual testing**.

Definition of Decision Table Analysis

Decision Table Analysis is a **black-box test design technique** used in manual testing to represent and analyze **different combinations of inputs and their corresponding expected outputs**.

It is especially useful when the system's behavior depends on a **set of rules, conditions, or combinations**. This helps ensure that all possible scenarios are tested.

Category It Falls Under

- **Test Design Technique**
 - **Black Box Testing**
 - Part of **Static and Dynamic Testing** (mostly dynamic)
 - Related to **Requirements-Based Testing**
-

Why Use Decision Table Analysis?

- To deal with complex business logic
 - To **reduce redundancy** in test cases
 - To ensure **coverage for all possible input combinations**
 - Helps in **identifying missing requirements** or rules
-

Real-Time Beginner Example

Scenario: Login System

Rules:

- If username and password are correct → Login Success
- If username is correct and password is incorrect → Error
- If username is incorrect and password is correct → Error
- If both are incorrect → Error

Rule	Username Correct	Password Correct	Output
1	Yes	Yes	Login Success
2	Yes	No	Error Message

3	No	Yes	Error Message
4	No	No	Error Message

Test Cases Derived:

1. Test valid username and password.
 2. Test valid username and invalid password.
 3. Test invalid username and valid password.
 4. Test both invalid username and password.
-

Intermediate Example

Scenario: ATM Withdrawal Rules

Rules:

- Account has sufficient balance
- Correct PIN entered
- Daily withdrawal limit not exceeded

Rule	Sufficient Balance	Correct PIN	Within Limit	Output
1	Yes	Yes	Yes	Withdrawal Success

2	Yes	Yes	No	Limit Exceeded
3	Yes	No	Yes	Incorrect PIN
4	No	Yes	Yes	Insufficient Funds
5	No	No	Yes	Incorrect PIN
6	No	Yes	No	Insufficient Funds
7	Yes	No	No	Incorrect PIN
8	No	No	No	Incorrect PIN

 Notice: Though some combinations like rule 6 or 7 are logically invalid, we still document them to **ensure complete test coverage**.

Advanced Example: Insurance Premium Calculator

Conditions:

- Age group: <25, 25-50, >50
- Smoker: Yes / No
- Health Issues: Yes / No

Expected Outcomes:

- High Premium

- Medium Premium
- Low Premium
- Rejected

Rule	Age Group	Smoker	Health Issues	Premium Decision
1	<25	Yes	Yes	High Premium
2	<25	Yes	No	Medium Premium
3	<25	No	No	Low Premium
4	25-50	Yes	Yes	High Premium
5	25-50	No	Yes	Medium Premium
6	25-50	No	No	Low Premium
7	>50	Yes	Yes	Rejected
8	>50	No	No	Medium Premium

 **Business Benefits:**

- Helps insurance domain testers cover various risk combinations.

- Prevents logical gaps in premium calculation logic.
-

Key Points to Remember

Element	Description
Conditions	Inputs or situations (Yes/No, True/False)
Actions/Outputs	System responses for combinations of conditions
Rules	Unique combinations of input values
Test Cases	Derived from each rule in the decision table

Tools to Create Decision Tables

- **Excel** (most common for manual testing)
 - **Test Management Tools** like TestRail, Zephyr
 - **Diagrams.net** or **Lucidchart** for visual representation
-

Would you like a template in Excel or a PDF for practice? Or should I guide you on how to draw these in a tool like Lucidchart or Draw.io?

State Transition Testing

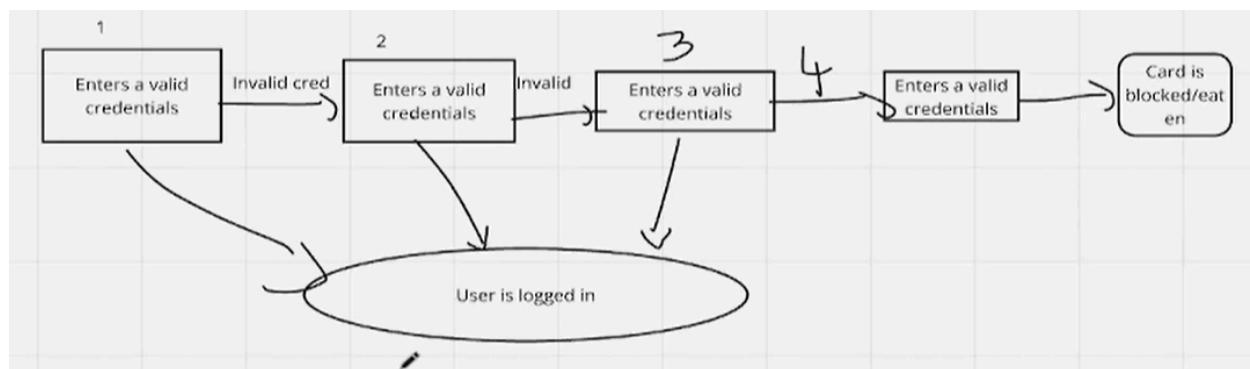
The state transition testing is conducted to verify the change in states of a software under varying inputs. If the circumstances under which the provided input are modified, then there are updates to the states of the software.

The software state transition testing comes under the black box testing and is performed to verify the characteristics of the software for multiple sets of input conditions that are fed to it in a specific order. It includes verification of both positive and negative flows. This type of testing is adopted where various state transitions of the software need to be verified.

Example

Let us take an example of a banking application where we would create a state transition diagram on login module features listed below –

- User enters correct credentials in the first attempt, user logs into the banking system.
- User enters correct credentials in the second attempt, user logs into the banking system.
- User enters correct credentials in the third attempt, user logs into the banking system.
- User enters correct credentials in the four attempts, user credentials are locked.



State Transition Testing

Certainly! Let's explore **State Transition Testing** thoroughly, starting from the **definition** to **real-time beginner to advanced examples**.



Definition of State Transition Testing

State Transition Testing is a **Black Box Testing** technique used to validate **different system states** and the **transitions** between those states based on specific inputs, events, or conditions.

In simpler terms:

You test how a system behaves when it **changes from one state to another**, like from "Logged Out" to "Logged In" based on valid credentials.



Category

- **Type:** Black Box Testing
 - **Belongs to:** Behavioral Testing Technique
 - **Used For:** Systems where **output depends on current state** and **input** (e.g., login/logout systems, ATMs, vending machines, traffic lights, etc.)
-



When to Use State Transition Testing

- When the system has **finite states** (e.g., ON/OFF, Approved/Rejected).
 - When the **output** depends not just on the input but also the **previous state**.
 - To find **missing transitions**, **invalid states**, or **unexpected behavior**.
-



Key Concepts

- **State:** A condition of the system at a point in time.
- **Transition:** A movement from one state to another, triggered by an event/input.

- **Event/Input:** The trigger that causes the transition.
 - **Action/Output:** Result of the transition.
-



Examples from Beginner to Advanced



Beginner Example 1: Login System



Scenario:

A user can log in with a correct password. After 3 incorrect attempts, the account is locked.



States:

- S1: Start
- S2: Logged In
- S3: Attempt 1
- S4: Attempt 2
- S5: Attempt 3
- S6: Locked



Transitions:

Current State	Input	Next State	Output
Start	Correct password	Logged In	Success
Start	Incorrect password	Attempt 1	Error Message
Attempt 1	Incorrect password	Attempt 2	Error Message
Attempt 2	Incorrect password	Attempt 3	Warning

Attempt 3	Incorrect password	Locked	Account Locked
-----------	--------------------	--------	----------------

You can draw a **state transition diagram** to visualize this.

🟡 Intermediate Example 2: Traffic Light Controller

📌 Scenario:

Traffic lights move in sequence: Red → Green → Yellow → Red

✓ States:

- Red
- Green
- Yellow

⟳ Transitions:

Current State	Event (Timer)	Next State
---------------	---------------	------------

Red	Timer	Green
Green	Timer	Yellow
Yellow	Timer	Red

This test ensures that no state skips occur, e.g., Red → Yellow (invalid).

🌐 Advanced Example 3: Online Order System

📌 Scenario:

Order goes through the following states:

Created → Paid → Shipped → Delivered → Returned/Cancelled

✓ States:

- Created
- Paid
- Shipped
- Delivered
- Returned
- Cancelled

Transitions:

Current State	Input/Event	Next State	Valid?
Created	Payment Received	Paid	Yes
Paid	Item Shipped	Shipped	Yes
Shipped	Item Delivered	Delivered	Yes
Delivered	Customer Returns	Returned	Yes
Delivered	Cancel	Cancelled	 No
Paid	Cancel	Cancelled	Yes

Test cases can verify:

- Invalid transitions (like Delivered → Cancelled)
- Proper transition sequences
- Handling of edge cases (e.g., cancelling after shipping)



Tips for Writing State Transition Test Cases

1. **Identify all states** of the system.

2. **List all events/inputs** that trigger transitions.
 3. **Draw a state transition diagram or table.**
 4. Write test cases for:
 - **Valid transitions**
 - **Invalid transitions**
 - **Edge states** (initial, terminal)
 5. Cover both **positive and negative scenarios.**
-

Summary

Aspect	Description
Technique Type	Black Box
Best For	Systems with multiple states and transitions
Common Use Cases	Login systems, order tracking, workflows, etc.
Test Focus	State + Input = Transition
Main Tool/Artifact	State Transition Table/Diagram

Would you like a **real-world state transition diagram** drawn or test case examples written in **template format** (like in Excel or table)?